




# The .NET Language Integrated Query Project

**PDC<sup>05</sup>**  
DEVELOPER POWERED

Anders Hejlsberg  
TLN306  
Technical Fellow  
Microsoft Corporation

**Microsoft®**

Problem:  
Data  $\neq$  Objects

Abstract blue and green brushstrokes or lines in the bottom right corner of the slide.

# The LINQ Project

FUTURE  
TECHNOLOGIES

C#

VB

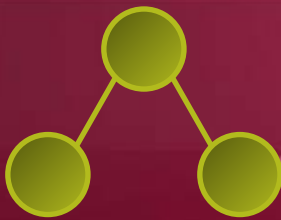
Others...

## .NET Language Integrated Query

Standard  
Query  
Operators

DLinq  
(ADO.NET)

XLinq  
(System.Xml)



Objects



SQL

WinF  
S

```
<book>
  <title/>
  <author/>
  <year/>
  <price/>
</book>
```

XML

Abstract colorful lines in pink, blue, and yellow, flowing from the left side of the slide.

# DEMO

## The LINQ Project

**PDC<sup>05</sup>**  
DEVELOPER POWERED

# Language Innovations

- Lambda expressions

```
c =>  
c.Name
```

- Extension methods

```
static void Dump(this  
object o);
```

- Local variable type inference

```
var x = 5;
```

- Object initializers

```
new Point { x = 1, y  
= 2 }
```

- Anonymous types

```
new { c.Name,  
c.Phone }
```

- Query expressions

```
from ... where ...  
select
```

- Expression trees

```
Expression<T  
>
```

# Standard Query Operators

Restriction	Where
Projection	Select, SelectMany
Ordering	OrderBy, ThenBy
Grouping	GroupBy
Quantifiers	Any, All
Partitioning	Take, Skip, TakeWhile, SkipWhile
Sets	Distinct, Union, Intersect, Except
Elements	First, FirstOrDefault, ElementAt
Aggregation	Count, Sum, Min, Max, Average
Conversion	ToArray, ToList, ToDictionary
Casting	OfType<T>

# Deferred Query Execution

```
Customer[] custs = SampleData.GetCustomers();
```

```
var query = from c in custs where c.City == "London" select
```

```
var query = custs.Where(c => c.City == "London").Select(c  
=> c.Name);
```

```
string[] names = query.ToArray();
```

custs

[illegible]

# Where

```
c ==> c.City == "London"
```

# Select

c =>  
c.Name

names

[illegible]



# DLinq For Relational Data

## Accessing data today

```
SqlConnection c = new SqlConnection(  
c.Open();  
SqlCommand cmd = new SqlCommand(  
    @"SELECT c.Name, c.Phone  
    FROM Customers c  
    WHERE c.City = @p0");  
cmd.Parameters.AddWithValue("@p0",  
"London");  
DataReader dr = c.Execute(cmd);  
while (dr.Read()) {  
    string name = dr.GetString(0);  
    string phone = dr.GetString(1);  
    DateTime date = dr.GetDateTime(2);  
}  
dr.Close();
```

Queries in  
quotes

Loosely  
bound  
arguments

Loosely  
typed result  
sets

No compile  
time checks



# DLinq For Relational Data

## Accessing data with DLinq

```
public class Customer { ... }
```

Classes  
describe data

```
public class Northwind: DataContext  
{  
    public Table<Customer> Customers  
    ...  
}
```

Tables are  
like  
collections

```
Northwind db = new Northwind(...);  
var contacts =  
    from c in db.Customers  
    where c.City == "London"  
    select new { c.Name, c.Phone };
```

Strongly  
typed  
connection

Integrated  
query syntax

Strongly  
typed results

# DLinq For Relational Data

- Language integrated data access
  - Maps tables and rows to classes and objects
  - Builds on ADO.NET and .NET Transactions
- Mapping
  - Encoded in attributes
  - Relationships map to properties
- Persistence
  - Automatic change tracking
  - Updates through SQL or stored

# XLinq For XML Data

## Programming XML today

```
XmlDocument doc = new XmlDocument();
XmlElement contacts = doc.CreateElement("contacts");
foreach (Customer c in customers)
    if (c.Country == "USA") {
        XmlElement e = doc.CreateElement("contact");
        XmlElement name = doc.CreateElement("name");
        name.InnerText = c.CompanyName;
        e.AppendChild(name);
        XmlElement phone = doc.CreateElement("phone");
        phone.InnerText = c.Phone;
        e.AppendChild(phone);
        contacts.AppendChild(e);
    }
doc.AppendChild(contacts);
```

Imperative  
model

Document  
centric

No integrated  
queries

Memory  
intensive

```
<contacts>
  <contact>
    <name>Great Lakes
Food</name>
    <phone>(503)
555-7123</phone>
  </contact>
  ...
</contacts>
```

# XLinq For XML Data

## Programming XML with XLinq

```
XElement contacts = new XElement("contacts",  
    from c in customers  
    where c.Country == "USA"  
    select new XElement("contact",  
        new XElement("name", c.CompanyName),  
        new XElement("phone", c.Phone)  
    )  
);
```

Declarative  
model

Element  
centric

Integrated  
queries

Smaller and  
faster

# XLinq For XML Data

- Language integrated query for XML
  - Expressive power of XPath / XQuery
  - But with C# or VB as programming language
- Leverages experience with DOM
  - Element centric, not document centric
  - Functional construction
  - Text nodes are just strings
  - Simplified XML namespace support
  - Faster and smaller

Abstract colorful lines in pink, blue, and yellow, flowing from the left side of the slide across a horizontal grey band.

# DEMO

## DLinq and XLinq

**PDC<sup>05</sup>**  
DEVELOPER POWERED



# The LINQ Project

- Language Integrated Query for .NET
  - Native query syntax in C# 3.0 and VB 9.0
- Standard Query Operators
  - SQL-like queries for any .NET collection
- DLinq
  - Query enabled data access framework
- XLinq
  - Query enabled, smaller, faster XML DOM



# Benefits Of LINQ

- Unified querying of objects, relational, XML
- Type checking and IntelliSense for queries
- SQL and XQuery-like power in C# and VB
- Extensibility model for languages /

# More Information

Wednesday	Thursday	Friday
<b>TLN306</b> - LINQ Overview 1:45 PM - 3:00 PM Halls C & D	<b>TLN308</b> - VB 9.0 10:00 AM - 11:15 AM Room 411	<b>DAT323 (R)</b> - DLinQ for SQL 8:30 AM - 9:45 AM Room 408 AB
<b>TLN307</b> - C# 3.0 3:15 PM - 4:30 PM Halls C & D	<b>TLN307 (R)</b> - C# 3.0 2:15 PM - 3:30 PM Room 402 AB	<b>DAT324 (R)</b> - XLinq 10:30 AM - 11:45 AM Room 408 AB
<b>DAT312</b> - DLinQ for WinFS 5:00 PM - 6:15 PM Room 515 AB	<b>DAT323</b> - DLinQ for SQL 2:15 PM - 3:30 PM Room 152/153 (Hall F)	<b>PNL11</b> - LINQ Panel 1:00 PM - 2:30 PM Room 152/153 (Hall F)
	<b>DAT324</b> - XLinq 5:15 PM - 6:30 PM Room 404	
	<b>Ask the Experts</b> 6:30 PM - 9:00 PM Main Hall	

<http://msdn.microsoft.com/netframework/future/linq/>



# ***Microsoft***<sup>®</sup>

*Your potential. Our passion.*<sup>™</sup>

© 2005 Microsoft Corporation. All rights reserved.

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.